

```
import pygame
import sys
import random

# Inicializar Pygame
pygame.init()

# Constantes
WIDTH, HEIGHT = 600, 400
BALL_RADIUS = 10
PADDLE_WIDTH, PADDLE_HEIGHT = 10, 60
FPS = 60
WHITE = (255, 255, 255)

# Configurar la pantalla
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Ping Pong")

# Reloj para controlar la velocidad de fotogramas
clock = pygame.time.Clock()

# Nuevas constantes para los niveles de dificultad
EASY = 1
MEDIUM = 2
HARD = 3

# Nivel de dificultad (ajustar según sea necesario)
difficulty = MEDIUM

# Velocidades iniciales
```

```

CPU_SPEED = 4
ball_speed_x = random.choice([-5, 5])
ball_speed_y = random.choice([-5, 5])
paddle_speed = 5

# Función para reiniciar la posición de la pelota
def reset_ball():
    global ball_x, ball_y, ball_speed_x, ball_speed_y
    ball_x, ball_y = WIDTH // 2, HEIGHT // 2
    ball_speed_x = random.choice([-5, 5])
    ball_speed_y = random.choice([-5, 5])

# Función para dibujar las paletas y la pelota
def draw_objects():
    screen.fill((0, 0, 0)) # Limpia la pantalla

    pygame.draw.rect(screen, WHITE, (5, paddle1_y, PADDLE_WIDTH,
PADDLE_HEIGHT)) # Paleta izquierda

    pygame.draw.rect(screen, WHITE, (WIDTH - PADDLE_WIDTH - 5, paddle2_y,
PADDLE_WIDTH, PADDLE_HEIGHT)) # Paleta derecha

    pygame.draw.circle(screen, WHITE, (ball_x, ball_y), BALL_RADIUS) # Pelota

    pygame.draw.aaline(screen, WHITE, (WIDTH // 2, 0), (WIDTH // 2, HEIGHT)) #
Línea central

# Bucle principal del juego
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    keys = pygame.key.get_pressed()

```

```

# Modo de un jugador
if keys[pygame.K_UP] and paddle1_y > 0:
    paddle1_y -= paddle_speed
if keys[pygame.K_DOWN] and paddle1_y < HEIGHT - PADDLE_HEIGHT:
    paddle1_y += paddle_speed

# Modo de dos jugadores
if keys[pygame.K_UP] and paddle2_y > 0:
    paddle2_y -= paddle_speed
if keys[pygame.K_DOWN] and paddle2_y < HEIGHT - PADDLE_HEIGHT:
    paddle2_y += paddle_speed

# Movimiento de la CPU (nivel de dificultad ajustable)
if ball_speed_x > 0:
    if difficulty == EASY:
        if paddle2_y + PADDLE_HEIGHT // 2 < ball_y and paddle2_y < HEIGHT - PADDLE_HEIGHT:
            paddle2_y += CPU_SPEED
        elif paddle2_y + PADDLE_HEIGHT // 2 > ball_y and paddle2_y > 0:
            paddle2_y -= CPU_SPEED
    elif difficulty == MEDIUM:
        if paddle2_y + PADDLE_HEIGHT // 2 < ball_y and paddle2_y < HEIGHT - PADDLE_HEIGHT:
            paddle2_y += CPU_SPEED // 2
        elif paddle2_y + PADDLE_HEIGHT // 2 > ball_y and paddle2_y > 0:
            paddle2_y -= CPU_SPEED // 2
    elif difficulty == HARD:
        if paddle2_y + PADDLE_HEIGHT // 2 < ball_y and paddle2_y < HEIGHT - PADDLE_HEIGHT:
            paddle2_y += CPU_SPEED * 2

```

```
elif paddle2_y + PADDLE_HEIGHT // 2 > ball_y and paddle2_y > 0:
    paddle2_y -= CPU_SPEED * 2

# Actualizar la posición de la pelota
ball_x += ball_speed_x
ball_y += ball_speed_y

# Resto del código sin cambios

# ... (resto del código)

# Verificar si la pelota salió por la izquierda o la derecha
if ball_x - BALL_RADIUS < 0:
    reset_ball()
    print("¡Puntuación para Jugador 2!")
elif ball_x + BALL_RADIUS > WIDTH:
    reset_ball()
    print("¡Puntuación para Jugador 1!")

# Dibujar objetos en la pantalla
draw_objects()

# Actualizar la pantalla
pygame.display.flip()

# Controlar la velocidad de fotogramas
clock.tick(FPS)
```